

Penerapan Algoritma A-Star Sebagai Pencari Rute Terpendek pada Robot Hexapod

Dedy Hermanto^{1*} dan Sepri Dermawan²

¹Jurusan Teknik Komputer, Akademi Manajemen Informatika dan Komputer MDP

²Teknik Informatika, Sekolah Tinggi Informatika dan Komputer GI MDP

*Corresponding author, e-mail: dedy@mdp.ac.id

Abstrak— Algoritma A-Star merupakan gabungan antara algoritma pencarian *Uniform Cost* dan *Greedy-Best First*. Algoritma *Uniform cost* digunakan untuk mencari biaya yang paling rendah untuk dimulai dari titik awal hingga akhir, sedangkan algoritma *Greedy-best first* digunakan untuk memberikan estimasi biaya dari titik awal hingga akhir dengan menggunakan fungsi heuristik. Salah satu implementasi dari algoritma A-Star digunakan untuk mencari jalur tercepat. Pada penelitian ini algoritma A-Star digunakan pada navigasi robot hexapod untuk mencari jalur tercepat. Hexapod robot merupakan salah satu jenis robot, dimana pergerakan robot ini dibantu dengan menggunakan enam kaki. Prinsip kerja dari robot dalam penelitian ini yaitu robot akan bergerak sesuai grid yang telah dipetakan dalam arena dengan memanfaatkan algoritma A-Star dan dengan bantuan dari sensor kompas dan ultrasonic. Sensor kompas digunakan untuk navigasi arah gerak robot dimana didapat dari algoritma A-Star. Sensor ultrasonic digunakan untuk mendeteksi halangan yang ada didepan, ketika terdapat halangan maka robot akan menghindar dan akan melakukan kembali perhitungan jalur tercepat lagi. Metodologi yang digunakan dalam penelitian ini dengan menggunakan metoda iteratif. Hasil yang didapat pada penelitian ini yang telah diuji sebelumnya didapat bahwa algoritma A-Star dapat digunakan pada robot hexapod untuk mendapatkan jalur tercepat dalam mencapai tujuan.

Kata Kunci : Algoritma A-Star, Robot Hexapod, Iteratif, Halangan dan Navigasi

Abstract— A-Star is a path finding algorithm that combine Uniform Cost Search and Greedy-Best First Search. Uniform Cost Search will choose the smallest cost from start to goal, while Greedy-Best First Search uses a heuristic function to estimate the cost required from start to goal. This research aims to implement A-Star algorithm to find the fastest path for hexapod robot navigation. Hexapod robot is a type of robot that moves using six legs. The working principle of hexapod robot in this research is the robot moves according grid obtained from A-Star algorithm with the help of compass sensors and ultrasonic sensors. The compass sensor is used to navigate the robot directions in path that obtained from A-Star, while ultrasonic sensors is used to detect any obstacle on the robot path. When ultrasonic sensors detect the obstacle, the robot will move away and find the fastest path again using A-Star. The method used in this research is the development of systems with an iterative model. Based on the results of the testing, the A-Star algorithm can be used to find the fastest path on hexapod robot.

Keywords : A-Star Algorithm, Hexapod Robot, Iterative, Obstacle and Navigation

1. Pendahuluan

Robot berasal dari kata “robota” yang dalam bahasa Ceko (*Czech*) berarti bekerja. Secara umum robot dapat diartikan sebagai mesin hasil rakitan manusia yang bekerja tanpa mengenal lelah [1]. Berdasarkan alat geraknya robot diklasifikasikan menjadi 2 (dua) jenis yaitu robot beroda dan robot berkaki. Robot beroda adalah robot yang dapat bermanuver dengan

menggunakan dua roda ataupun lebih. Robot berkaki adalah robot yang bergerak dengan kaki-kaki buatan. Robot berkaki yang umum ditemui salah satunya adalah robot berkaki enam atau sering disebut dengan robot *hexapod*. Robot *hexapod* memiliki fleksibilitas yang tinggi karena robot secara statistik dapat stabil dengan menggunakan tiga kaki [2].

Untuk menyelesaikan pekerjaannya, robot harus dapat bergerak secara *autonomous*.

Received date 2018-4-27, Revised date 2018-5-29, Accepted date 2018-7-11

<https://doi.org/10.25077/jnte.v7n2.545.2018>

This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

Sebuah persyaratan utama bagi sebuah robot yang benar-benar *autonomous* adalah dapat bernavigasi dengan aman pada lingkungannya. Untuk bernavigasi dengan aman, robot dapat menggunakan pencarian jalur (*pathfinding*). *Goal* dari *pathfinding* ini pada umumnya adalah untuk mencari jalur tercepat dengan sebisa mungkin menghindari penghalang (*obstacle*) yang ada. Adapun penghalang dapat berupa dinding dan sebagainya. Salah satu metode *pathfinding* yang dapat digunakan pada robot adalah algoritma A*.

Algoritma A* (dibaca “A STAR”) adalah algoritma *Best First Search* yang merupakan perpaduan antara *Uniform Cost Search* yang memilih jarak paling kecil dari simpul awal ke simpul berikutnya dan *Greedy-Best First Search* yang menggunakan heuristik atau nilai perkiraan untuk menentukan simpul berikutnya. Algoritma A* ini akan menemukan rute yang *complete* (selalu menemukan solusi (jika memang ada solusinya)) dan optimal [3].

Pada penelitian sebelumnya, penerapan algoritma A* dilakukan pada robot beroda. Penelitian ini dilakukan oleh Rahul Sharma K pada tahun 2013 dengan judul *design and implementation of path planning algoritm for wheeled mobile robot in a known dynamic environment*. Hasil menunjukkan bahwa algoritma A* efektif untuk menghitung jalur terpendek pada lingkungan dinamis yang sudah diketahui dan memungkinkan robot untuk cepat menyelesaikan misi [4]. Pada penelitian ini dilakukan penerapan algoritma A* pada robot *hexapod* guna mencari jalur tercepat. Gambar 1 menunjukkan bentuk dari jenis robot hexapod yang akan dibuat.

2. Tinjauan Pustaka

Robot *hexapod* merupakan salah satu jenis robot yang bermanuver menggunakan kaki sebanyak 6 buah. Untuk mencapai tujuan dengan cepat, digunakan algoritma A* sebagai pencari jalur terbaik. Robot juga dibantu dengan sensor kompas dan ultrasonik untuk melakukan navigasi.

2.1 Robot Hexapod

Sesuai dengan namanya, robot *hexapod* adalah jenis *mobile* robot yang bergerak

menggunakan 6 (enam) kaki. Karena robot secara statistik dapat stabil dengan menggunakan 3 (tiga) kaki, maka robot *hexapod* tentu memiliki fleksibilitas yang lebih tinggi [5].



Gambar 1. Robot *Hexapod*

2.2 Algoritma A-Star

Algoritma A* (dibaca “A STAR”) adalah algoritma *Best First Search* yang merupakan perpaduan antara *Uniform Cost Search* dan *Greedy-Best First Search*. *Uniform Cost Search* ini akan memilih jarak paling kecil dari simpul awal ke simpul berikutnya sampai ke simpul tujuan, sedangkan *Greedy-Best First Search* yang menggunakan fungsi heuristik akan memperkirakan biaya dari simpul awal ke simpul tujuan. Heuristik ini memiliki peran yang sangat penting untuk mengontrol pencarian pada algoritma A*, sehingga algoritma ini akan menemukan rute yang *complete* (selalu menemukan solusi (jika memang ada solusinya)) dan optimal [6].

Prinsip dari algoritma A* adalah melakukan traversal satu per satu pada tiap simpul untuk memperoleh lintasan terpendek. Algoritma ini akan menghitung jarak salah satu lintasan, lalu menyimpannya dan kemudian menghitung jarak lintasan lainnya. Ketika seluruh lintasan telah selesai dihitung, algoritma A* akan memilih lintasan yang paling pendek. Rumus yang digunakan pada algoritma A* adalah :

$$F(n) = G(n) + H(n) \quad (1)$$

dimana,

$F(n)$ = biaya yang dibutuhkan,

$G(n)$ = biaya yang ditempuh dari node asal,

$H(n)$ = nilai perkiraan dari node saat ini ke tujuan.

3. Metoda Penelitian

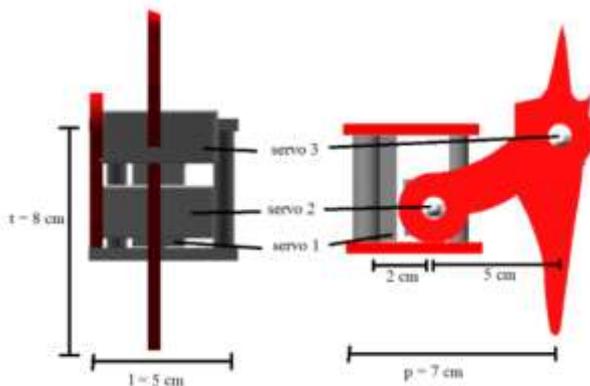
Penelitian ini dilakukan dalam 2 tahap yaitu merancang perangkat keras dan kemudian melakukan perancangan terhadap perangkat lunak yang akan digunakan. Perangkat keras yang dirancang berupa robot hexapod yang akan diimplementasikan ke algoritma A*, agar dapat menjadi jalan terbaik untuk mencapai tujuan.

3.1 Perancangan Perangkat Keras Robot Hexapod

Penelitian menggunakan robot hexapod, terlebih dahulu dilakukan perancangan badan/body dari robot yang akan digunakan kemudian dilanjutkan dengan perangkat keras yang dipakai.

a. Rancangan Kaki Robot

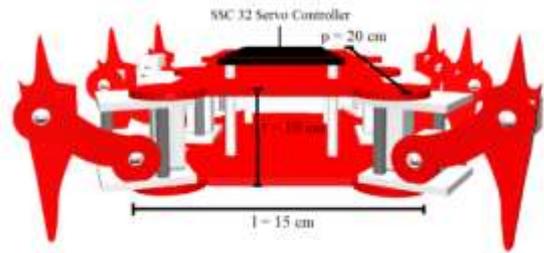
Rancangan bagian kaki robot hexapod yang digunakan pada penelitian ini tersaji pada Gambar 2. Pada rancangan tersebut menunjukkan bahwa desain yang direncanakan yaitu menggunakan 3 buah aktuator tiap kaki, sehingga total aktuator/servo yang akan digunakan sebanyak 18 (delapan belas).



Gambar 2. Rancangan Kaki Robot Hexapod

b. Rancangan Body Robot

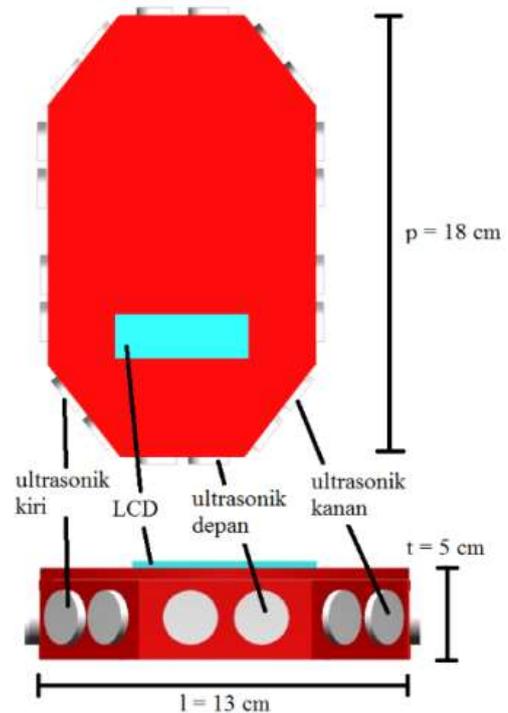
Setelah rancangan kaki robot dibuat, langkah berikutnya yaitu melakukan desain body robot. Gambar 3, menunjukkan desain body robot yang telah terhubung dengan kaki yang sebelumnya dirancang. Tujuan utama dibuat body ini yaitu untuk menggabungkan antara kaki satu dengan lainnya, juga untuk meletakkan komponen perangkat keras lainnya seperti controller.



Gambar 3. Rancangan Body Robot Hexapod

c. Rancangan Penutup Robot

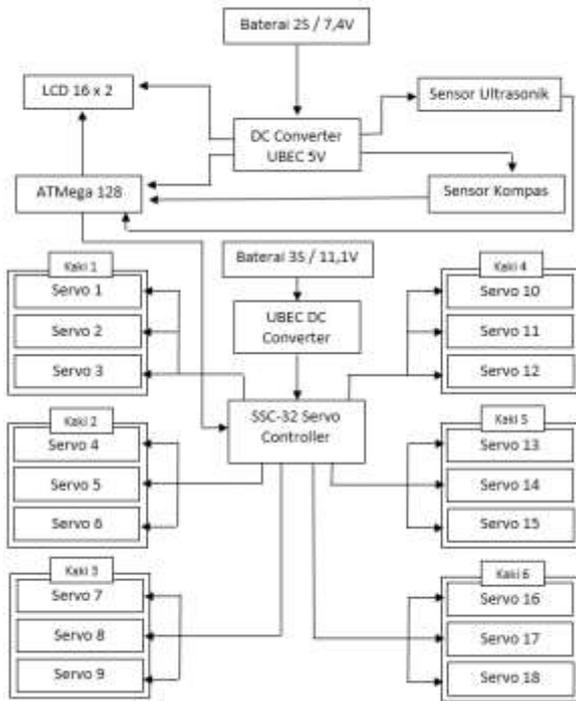
Setelah kaki dan body robot selesai desain, tahap selanjutnya yaitu membuat desain penutup kepala robot. Desain ini tersaji pada Gambar 4, tujuan utama dibuat bagian ini yaitu untuk tempat meletakkan perangkat keras sensor dan LCD.



Gambar 4. Rancangan Penutup Kepala Robot Hexapod

d. Blok Diagram Perangkat Keras

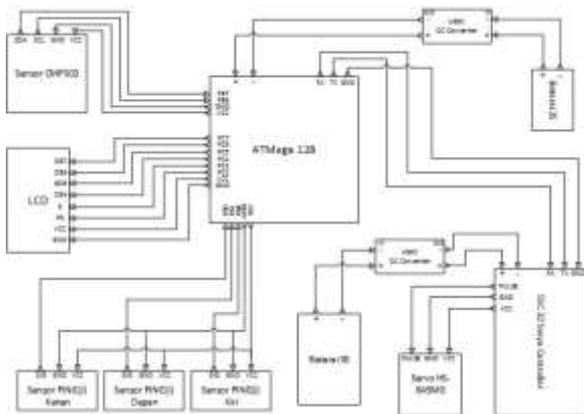
Perancangan body robot telah selesai, selanjutnya merancang peralatan apa saja yang dibutuhkan untuk dapat membuat robot hexapod. Dalam penelitian kali ini peneliti menggunakan peralatan sesuai dengan Gambar 5, dimana terdapat 2 sensor yaitu kompas dan ultrasonik dan dibantu dengan beberapa aktuator.



Gambar 5. Diagram Blok Perangkat Keras

e. Hubungan Perangkat Keras dan Controller

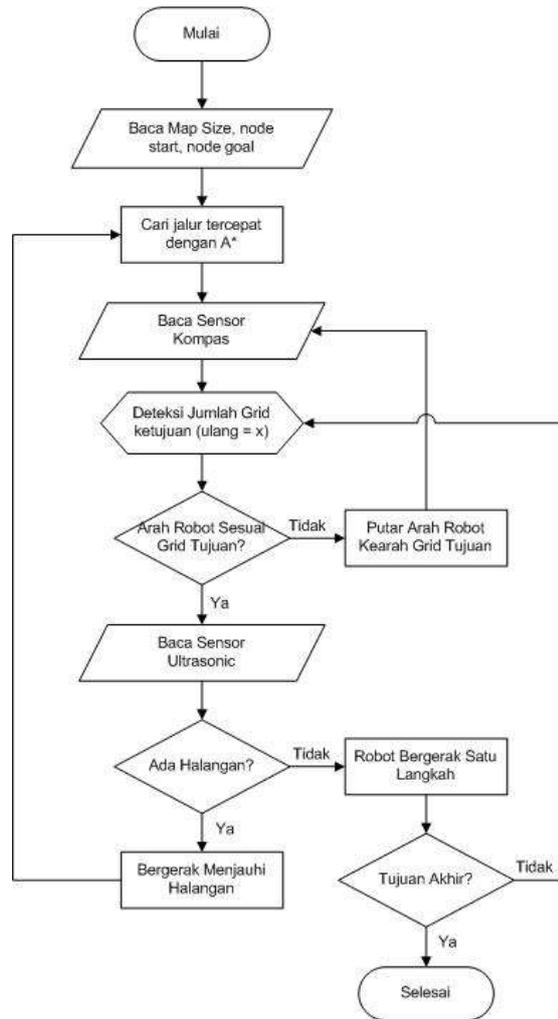
Agar dapat memudahkan pemahaman perangkat controller terhubung ke aktuator dan sensor terhubung ke bagian tertentu dari controller, maka dari Gambar 6 ditunjukkan hubungan tiap perangkat.



Gambar 6. Hubungan Perangkat Keras dan Controller

3.2 Perancangan Perangkat Lunak

Proses perancangan perangkat lunak yang dilakukan untuk penelitian ini yaitu tersaji pada Gambar 7.

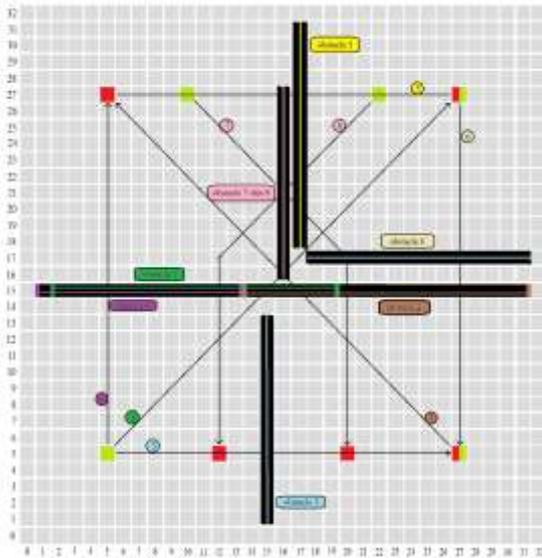


Gambar 7. Sistem Cara Kerja Robot Hexapod

Pada proses sistem kerja yang tersaji pada Gambar 7 yaitu ukuran arena, posisi mulai, node dan tujuan telah ditentukan terlebih dahulu. kemudian proses pencarian jalur dilakukan dengan menggunakan algoritma A* yang telah tertanam. Proses berjalan sesuai dengan node yang telah ditentukan, hingga robot hexapod mencapai tujuan yang akan dicapai.

3.3 Perancangan Arena Implementasi

Pada penelitian penerapan algoritma A* ini, peneliti melakukan perancangan arena yang akan digunakan untuk menerapkan algoritma A* pada robot Hexapod. Gambar 8 menunjukkan rancangan dari arena yang akan digunakan. Setiap grid memiliki sisi sebesar 4,5 cm.

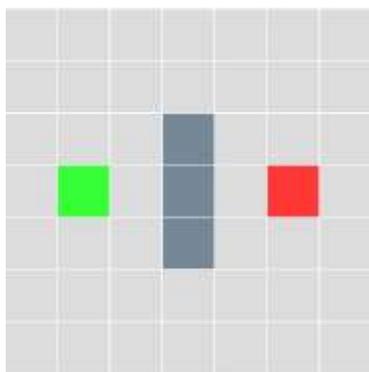


Gambar 8. Konfigurasi Posisi *Start*, *Goal* dan *Obstacle*

Pada Gambar 8 terdapat delapan pengujian yakni pada posisi *start* dan *goal* secara berturut-turut (5,5 ; 27,27), (27,5 ; 5,27), (5,5 ; 27,5), (5,5 ; 5,27), (27,27 ; 5,27), (27,27 ; 27,5), (10,27 ; 20,5), (22,27 ; 12,5). *Obstacle* adalah penghalang berupa dinding yang menghalangi jalur robot *hexapod* yang didapat dari algoritma A*. Jumlah *obstacle* yang digunakan adalah satu dan penempatannya dapat berubah-ubah sesuai dengan konfigurasi yang ada.

3.4 Perancangan Penerapan A*

Proses pencarian jalur tercepat yang dilakukan pada penelitian ini dengan menggunakan algoritma A*, dicontohkan pada Gambar 9.

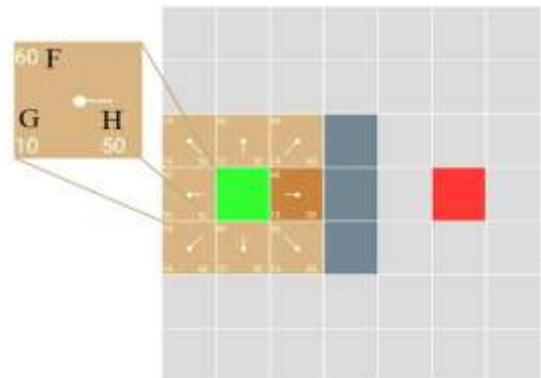


Gambar 9. Konfigurasi Posisi *Start*, *Goal* dan *Obstacle*

Pada Gambar 9 menunjukkan bahwa keadaan awal dimana kotak berwarna hijau adalah posisi *start* robot, kotak berwarna merah adalah posisi *goal* robot, dan kotak berwarna hitam adalah *obstacle* yang tidak boleh dilalui oleh robot. Robot dapat melangkah ke arah vertikal, horisontal, dan diagonal. Setiap langkah yang dilakukan robot disimbolkan dengan G. Nilai G *cost* untuk pergerakan vertikal dan horisontal adalah 10 sedangkan pergerakan diagonal diberi nilai 14. Nilai 14 didapat dari perhitungan *pythagoras*

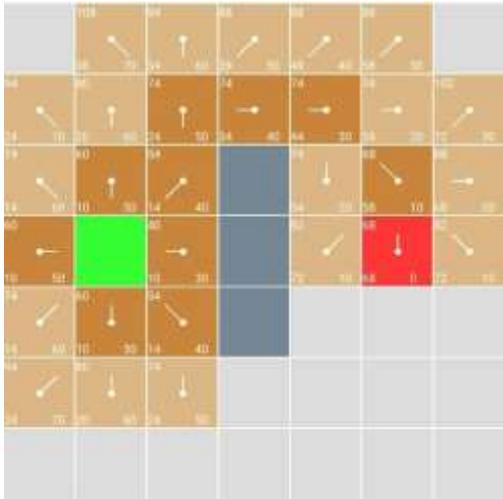
$$\sqrt{10^2 + 10^2} = 14,14213 \quad (2)$$

Nilai heuristik yang disimbolkan dengan H adalah 10 untuk setiap satu grid. Untuk mencari jalur tercepat A* akan menjumlahkan nilai G dan H pada setiap grid yang disimbolkan dengan F. Pemberian nilai F, G, dan H dapat dilihat pada Gambar 10.



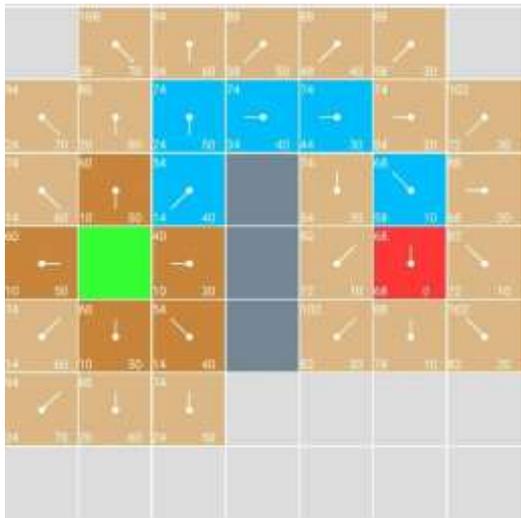
Gambar 9. *Transversal* ke *Neighbour* terdekat

A* akan melakukan *traversal* ke *neighbour* terdekat dan menyimpannya pada *open* (kotak berwarna coklat muda pada Gambar 9). Setelah semua *neighbour* terdekat dikunjungi, A* akan memilih grid dengan nilai F terkecil dan menyimpannya pada *closed* (kotak berwarna coklat tua pada Gambar 9). Langkah tersebut akan terus diulang sampai grid *goal* berada pada *closed* yang dapat dilihat pada Gambar 10.



Gambar 10. *Transversal Up to Grid Goal*

Setelah grid goal berada pada closed, A* akan mencari jalur tercepat dengan melakukan *backtracking* sampai ke grid start. Jalur tersebut dapat dilihat pada Gambar 11 yang ditandai dengan kotak berwarna biru.



Gambar 11. Jalur Tercepat

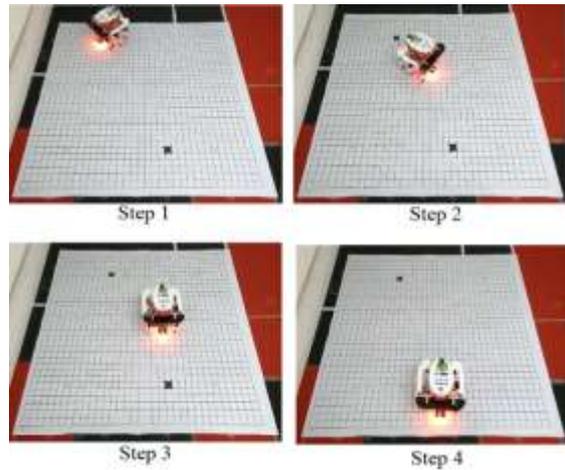
4. Hasil dan Pembahasan

Proses pengujian hasil penelitian dilakukan pengujian kesesuaian dengan jalur yang dihitung secara manual, ketepatan mencapai tujuan dan kecepatan mencapai tujuan.

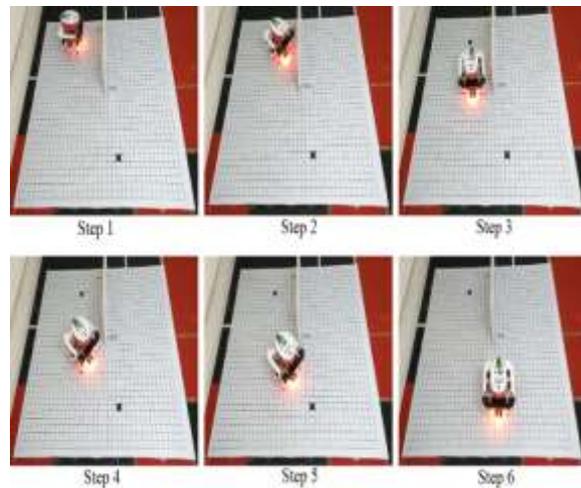
4.1. Pengujian Ketepatan Mencapai Tujuan

Pada Gambar 12 menunjukkan hasil pengujian pergerakan robot tanpa halangan sesuai dalam mencapai tujuan yang ingin dicapai

dan Gambar 13 menunjukkan hasil dimana robot mencapai tujuan dengan melewati suatu halangan yang sebelumnya telah terpasang.



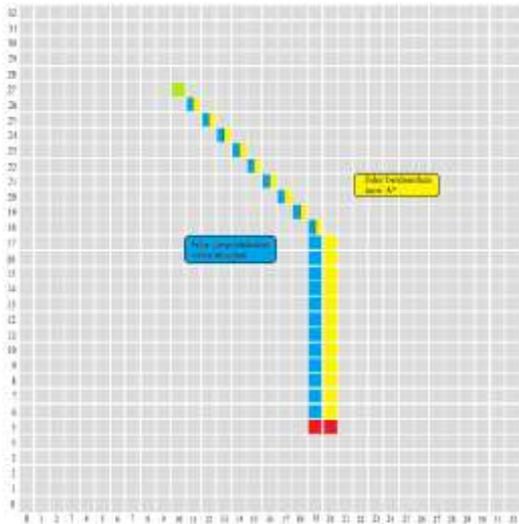
Gambar 12. Pergerakan Tanpa Halangan



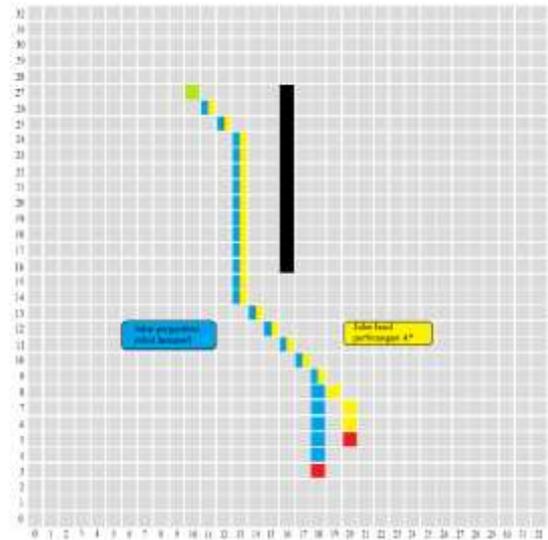
Gambar 13. Peregerakan Dengan Halangan

4.2. Pengujian Perbandingan Dengan Jalur Hasil Perhitungan

Pengujian selanjutnya yang dilakukan yaitu melakukan pengujian berupa perbandingan jalur yang merupakan hasil perhitungan menggunakan algoritma A* dan penerapan Algoritma A* yang terdapat dalam robot Hexapod. Dalam pengujian Gambar 14 dan Gambar 15, diwakilkan dengan menggunakan 2 warna yang berbeda, warna kuning untuk menggambarkan pergerakan robot hasil perhitungan manual sebelum diimplementasi kedalam robot dan warna biru pergerakan setelah dimasukkan kedalam robot.



Gambar 14. Pergerakan Robot Tanpa Halangan



Gambar 15. Pergerakan Robot Dengan Halangan

Dari hasil pengujian berdasarkan akurasi gerak robot dan menuju titik akhir didapat hasil sesuai dengan Tabel 1.

Dari Tabel 1, menunjukkan hasil ujicoba ketepatan ketika robot mencapai titik tujuan. Robot mulai berjalan pada titik (10,27) dan tujuan yang ingin dicapai (20,5), untuk titik

Tabel 1. Pengujian Akurasi Ketepatan dan Waktu Robot Mencapai Tujuan

Test	A	B	C	D	E	F
1	18,3	-2,-2	31,5	18,3	-2,-2	36,2
2	19,6	-1,1	29,9	17,5	-3,0	33,0
3	20,4	0,-1	32,7	16,5	-4,0	34,8
4	21,6	1,1	30,2	17,3	-3,-2	35,0
5	22,5	2,0	30,1	17,4	-3,-1	33,8
6	21,6	1,1	30,5	17,3	-3,-2	34,8
7	20,5	0,0	30,4	17,3	-3,-2	35,3
8	21,3	1,-2	33,3	17,2	-3,-3	33,7
9	19,5	-1,0	30,3	17,3	-3,-2	36,7
10	19,6	-1,1	30,3	16,2	-4,-3	32,6
Total		X=10, Y= 9	309,2		X=31, Y= 17	345,9

Keterangan:

- A: Pengujian Tanpa Halangan (grid(x,y))
- B: Error Posisi Tujuan (grid(x,y))
- C: Capaian Waktu Tanpa Halangan (s)

- D: Pengujian dengan Halangan (grid(x,y))
- E: Error Posisi Tujuan (grid(x,y))
- F: Capaian Waktu Dengan Halangan (s)

mulai dan titik akhir ini telah ditentukan terlebih dahulu, yang akan digunakan sebagai masukan untuk robot menentukan jalur terbaik. Rata-rata grid (*Average Grid*) yang didapat dari kesalahan mencapai tujuan ketika tanpa menggunakan halangan yaitu:

$$X = \frac{\text{AverageGrid}X}{\text{TotalTest}} = \frac{10}{10} = 1 \text{ grid} \quad (3)$$

$$Y = \frac{\text{AverageGrid}Y}{\text{TotalTest}} = \frac{9}{10} = 0,9 \text{ grid} \quad (4)$$

Kemudian pengujian berikutnya menggunakan halangan, dimana rata-rata yang didapat yaitu:

$$X = \frac{\text{AverageGrid}X}{\text{TotalTest}} = \frac{31}{10} = 3,1 \text{ grid} \quad (5)$$

$$Y = \frac{\text{AverageGrid}Y}{\text{TotalTest}} = \frac{17}{10} = 1,7 \text{ grid} \quad (6)$$

Dari hasil yang didapat yaitu, ketika tanpa menggunakan halangan, robot ketika mencapai titik tujuan hanya bergeser 1 grid. Ketika menggunakan halangan rata-rata pergeseran yaitu 2-3 grid.

Robot yang telah ditanamkan algoritma A*, dilakukan juga pengujian lamanya waktu dalam mencapai tujuan mulai dari titik awal sampai ke titik akhir. Pada Tabel 1 menunjukkan waktu hasil robot hexapod berjalan dari titik awal menuju titik akhir dengan menggunakan halangan dan tanpa halangan.

Dari Tabel 1, menunjukkan hasil kecepatan rata-rata robot hexapod untuk berjalan dari titik mulai yaitu (10,27) menuju titik akhir (20,5). Hasil rata-rata pertama ini diambil dari data tanpa halangan.

$$\text{AverageSped} = \frac{\text{TotalSpeed}}{\text{TotalTest}} = \frac{309.2}{10} = 30.92s \quad (7)$$

Ketika menggunakan halangan kecepatan rata-rata yang didapat yaitu:

$$\text{AverageSped} = \frac{\text{TotalSpeed}}{\text{TotalTest}} = \frac{345.9}{10} = 34.59s \quad (8)$$

5. Kesimpulan

Berdasar hasil pengujian yang telah dilakukan dalam penelitian ini dimana algoritma A* dapat diterapkan dalam robot hexapod untuk dapat membantu menemukan jalur terpendek mencapai titik tujuan, tingkat kesuksesan robot dapat sampai ketujuan yaitu dari langkah robot tidak terhadang oleh hambatan dari arena yang

digunakan dan hasil rata-rata capaian robot hexapod untuk proses ke titik tujuan tingkat kesalahan paling rendah 1 grid dan paling besar 3 grid.

Daftar Pustaka

- [1] W. Budhiarto, *Perancangan dan Pemrograman Hasta Karya Robot*, Andi, Yogyakarta, 2014.
- [2] Darwison and R. Wahyudi, "Kontrol Kecepatan Robot Hexapod Pemadam Api menggunakan Metoda Logika Fuzzy", dari Jurnal Nasional Teknik Elektro, 2015, h. 227-234.
- [3] A.S. Rosa & S. Muhammad, *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*, Informatika, Bandung, 2013.
- [4] W. Budiharto, *Belajar Sendiri Membuat Robot Cerdas*, Elex Media Komputindo, Jakarta, 2006.
- [5] J.W. Kusuma dkk, *Penerapan Invers Kinematik Terhadap Pergerakan Kaki pada Robot Hexapod*, STMIK GI MDP (10 Agustus 2016), dari <http://www.eprints.mdp.ac.id>.
- [6] P.J Juliantono dkk, *Implementasi dan Analisis Algoritma A*(Star) untuk Menentukan Jalur Dengan Multiple Goal pada Pergerakan NPC (Non-Playable Character)*, Universitas Telkom (13 Agustus 2016), <https://openlibrary.telkomuniversity.ac.id>.

Biodata Penulis

Dedy Hermanto, Menyelesaikan D3 tahun 2005 Jurusan Teknik komputer AMIK MDP. S1 jurusan Teknik Informatika STMIK GI MDP diselesaikan tahun 2009 Kemudian melanjutkan ke program pascasarjana di Universitas Indonesia Fakultas Ilmu Komputer Jurusan Teknologi Informasi. Saat ini sebagai dosen pada Akademi Manajemen Informatika dan Komputer MDP jurusan Teknik Komputer.

Sepri Dermawan, mendapat gelar sarjana pada tahun 2017 pada jurusan Teknik Informatika STMIK GI MDP dan saat ini bekerja sebagai programmer pada perusahaan jasa.